DeformPAM: Data-Efficient Learning for Long-horizon Deformable Object Manipulation via Preference-based Action Alignment

Wendi Chen[12*], Han Xue[12*], Fangyuan Zhou[1], Yuan Fang[1] and Cewu Lu[13]

***Abstract*— In recent years, imitation learning has made progress in the field of robotic manipulation. However, it still faces challenges when addressing complex long-horizon tasks with deformable objects, such as high-dimensional state spaces, complex dynamics, and multimodal action distributions. Traditional imitation learning methods often require a large amount of data and encounter distributional shifts and accumulative errors in these tasks. To address these issues, we propose a data-efficient general learning framework (DeformPAM) based on preference learning and reward-guided action selection. DeformPAM decomposes long-horizon tasks into multiple action primitives, utilizes 3D point cloud inputs and diffusion models to model action distributions, and trains an implicit reward model using human preference data. During the inference phase, the reward model scores multiple candidate actions, selecting the optimal action for execution, thereby reducing the occurrence of anomalous actions and improving task completion quality. Experiments conducted on three challenging real-world long-horizon deformable object manipulation tasks demonstrate the effectiveness of this method. Results show that DeformPAM improves both task completion quality and efficiency compared to baseline methods even with limited data. Code and data will be available at deform-pam.robotflow.ai.**

I. Introduction

Efficiently learning to perform general manipulation tasks has been a persistent focus in the field of robotics. In recent years, imitation learning has made significant progress in robotic manipulation [1–5]. However, these algorithms usually require a huge amount of data (*e.g.*, thousands of demonstrations) for complex long-horizon deformable object manipulation tasks [5]. Such tasks present the following unique properties:

- **High-dimensional state space** that often leads to complex initial and intermediate object states.
- **Complex dynamics** that are difficult to accurately model in simulations.
- **Multi-modal distribution** in action space.

These characteristics lead to significant distribution shifts and accumulation errors in traditional imitation learning algorithms for complex long-horizon tasks [6]. For an action policy modeled with probabilistic models (e.g., diffusion [1]), encountering unseen complex states makes the agent gradually drift away from the desired trajectory (see Fig. 1 left). To ensure that the data covers the high-dimensional state space as much as possible, the cost of collecting data in the real world will increase significantly. So, how can we learn to perform complex long-horizon deformable object manipulation tasks with a limited amount of data?

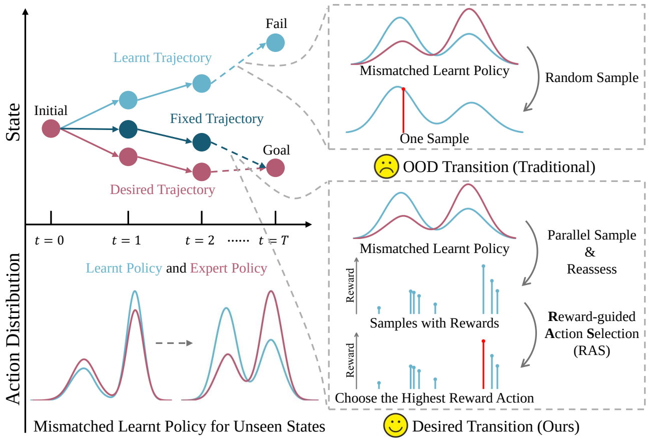


Fig. 1: In long-horizon manipulation tasks, a probabilistic policy may encounter distribution shifts when imperfect policy fitting leads to unseen states. As time progresses, the deviation from the expert policy becomes more significant. Our framework employs **R**eward-guided **A**ction **S**election (**RAS**) to reassess sampled actions from the generative policy model, thereby improving overall performance.

Our core idea is simple: we try to make the policy model distinguish between *good* and *bad* actions and only select the best action (see Fig. 1 right) during inference. This will reduce abnormal actions and alleviate distribution shifts in long-horizon tasks. Reward functions are a common way to evaluate actions, but as in previous works [7, 8], designing a reward function for each task individually has much hidden cost. Therefore, we choose to use human preference data as a general representation across tasks for evaluating action quality.

Based on this idea, we propose a general learning framework **DeformPAM** (see Fig. 2) for long-horizon **Deform**able object manipulation via **P**reference-based **A**ction align**M**ent. Our approach has three stages: **(1)** In the **first stage**, we collect a small amount of human demonstration data and use supervised learning to train an initial probabilistic policy model based on diffusion [9] and action primitives. **(2)** In the **second stage**, we run rollouts on real robots with the initial probabilistic policy model and record the N predicted actions for each state, which are then annotated with preference data by humans. We use DPO (Direct Preference Optimization) [10] on diffusion models [11] to directly learn an implicit reward model from this preference data. **(3)** Finally, during **inference**, we use **R**eward-guided **A**ction **S**election (**RAS**) to boost the performance of the initial policy model from the first stage. Specifically, we use the initial policy model

[1]Shanghai Jiao Tong University. [2]Meta Robotics Institute, SJTU. [3]Shanghai Innovation Institute. * indicates equal contribution. {chenwendi-andy, xiaoxiaoxh, ui-micro, sjtu_fy, lucewu}@sjtu.edu.cn

to generate N actions, score them using the implicit reward model, and select the action with the highest reward score to execute.

We find that this approach effectively reduces the occurrence of anomalous actions, thereby improving the performance of complex long-horizon tasks for deformable objects.

To validate the effectiveness of our learning framework, we conducted extensive real-world experiments on three challenging long-horizon deformable object manipulation tasks involving granular (granular pile shaping), 1D (rope shaping), and 2D (T-shirt unfolding) deformable objects. All these tasks start with very complex initial object states. We use IoU, coverage, and Earth Mover's Distance (EMD) to quantitatively measure the task completion quality of the model. Real-world results indicate that our method improves task completion quality and time compared to baseline methods across multiple complex tasks. Our contributions are summarized as follows:

- We design a general primitive learning framework (DeformPAM) for long-horizon deformable object manipulation, which uses an implicit reward model trained by preference data to select the action with higher quality.
- We evaluate our method with real robots on several highly challenging long-horizon deformable object manipulation tasks.

II. Related Works

A. *Deformable Object Manipulation*

Deformable object manipulation is a field with a long research history and numerous applications. Most methods in this domain typically construct specific simulation environments tailored to particular object types [12–15], designing specialized rewards [7, 8] or learning pipelines [16, 17] to accomplish specific tasks. These hidden costs make it challenging for these learning frameworks to generalize across tasks. Recently, Differentiable Particles [18] attempted to use a differentiable simulator to plan optimal action trajectories applicable to various tasks. However, it requires additional object state estimators as input, whereas our approach learns actions directly from raw point clouds. AdaptiveGraph [19] is a model-based method for general-purpose deformable object manipulation, which learns the dynamics model of deformable objects using massive data in simulation and online interaction data in the real world, followed by using MPC to plan optimal execution trajectories. However, like Differentiable Particles [18], this method requires building simulation environments for each object type and each task, and it also suffers from the sim-to-real gap due to complex dynamics of deformable objects.

B. *Imitation Learning for Long-horizon Manipulation*

In recent years, there have been two main approaches to extend imitation learning to complex long horizon tasks: hierarchical imitation learning [20–24] and learning from play data [25–28]. Hierarchical imitation learning decomposes task learning into high-level planning and low-level controllers, while the latter approach collects interaction environment data through human teleoperation of robotic arms, without requiring specific task goals. Our method is more akin to hierarchical imitation learning, which improves sample efficiency by utilizing atomic action skills. However, these learning methods usually perform experiments on long-horizon tasks with rigid objects [24, 28], or assume simple initial object states [25] (*e.g.*, flattened cloth). In comparison, our framework focuses on long-horizon tasks with deformable objects in complex initial states. RoboCook [29] is a framework for learning long horizon tasks involving deformable objects, but it is specifically designed for elasto-plastic objects (*i.e.*, dough), making it difficult to adapt directly to 1D (e.g., ropes) and 2D (e.g., garments) deformable objects. In contrast, our method theoretically applies to deformable objects of various dimensions.

C. *Learning from Human Preference*

Learning from human preference data [30–32] has garnered attention in the field of robotics. Recently, reinforcement learning from human feedback (RLHF) [33, 34] has become a popular way of leveraging preference data for aligning policy models (*e.g.*, large language models). Subsequently, to eliminate the reliance on an explicit reward model in RLHF, DPO [10] and CPL [35] enable direct policy finetuning from preference data, based on contextual bandits and Markov decision processes respectively. Additionally, PFM [36] learns a conditional flow matching model from preference data to optimize the actions predicted by the policy model. Owing to their convenience, this methodology has also been applied in fields like image generation (Diffusion-DPO [11]). Instead of directly using the finetuned policy model [11] or learning an action transformation model [36], we leverage the underlying implicit reward model of DPO to guide action selection from multiple generated action samples, which has been proven to be beneficial in natural language processing (NLP) [37, 38].

III. Preliminary

A. *Conditional Diffusion Models*

Diffusion models are a series of generative models that excel at generating samples x_0 from arbitrary multimodal distributions by progressively denoising Gaussian noise x_T. They can be conditional when given some condition c. A conditional diffusion model comprises two processes: the forward diffusion process and the reverse denoising process. They are considered as a Markov chain with fixed transitions q and learnable transitions p_θ respectively, which can be expressed as

$$q(x_t|x_{t-1}) : x_t = \alpha_t^{1/2} x_{t-1} + (1-\alpha_t)^{1/2}\epsilon_{t-1}, \tag{1}$$

$$p_\theta(x_{t-1}|x_t, c) : x_{t-1} = \mu_\theta(x_t, c, t) + (\Sigma_\theta(x_t, c, t))^{1/2}\xi_{t-1}. \tag{2}$$

where $\{\alpha_t \in (0,1)\}_1^T$ are the predefined variance schedule and ϵ, ξ are Gaussian noise. Moreover, an expression for directly calculating the diffusion result can be written as

$$q(x_t|x_0) : x_t = \prod_{i=1}^{t} \alpha_i^{1/2} x_0 + (1 - \prod_{i=1}^{t} \alpha_i)^{1/2}\epsilon. \tag{3}$$

During training, reparameterize μ_θ as $\mu_\theta(\epsilon_\theta, x_0)$ with Eq. 3 and a simplified ELBO objective in DDPM [9] is derived as

$$L_{simple} = \mathbb{E}_{x_0,t,\epsilon}\|\epsilon - \epsilon_\theta(x_t, c, t)\|_2^2. \tag{4}$$

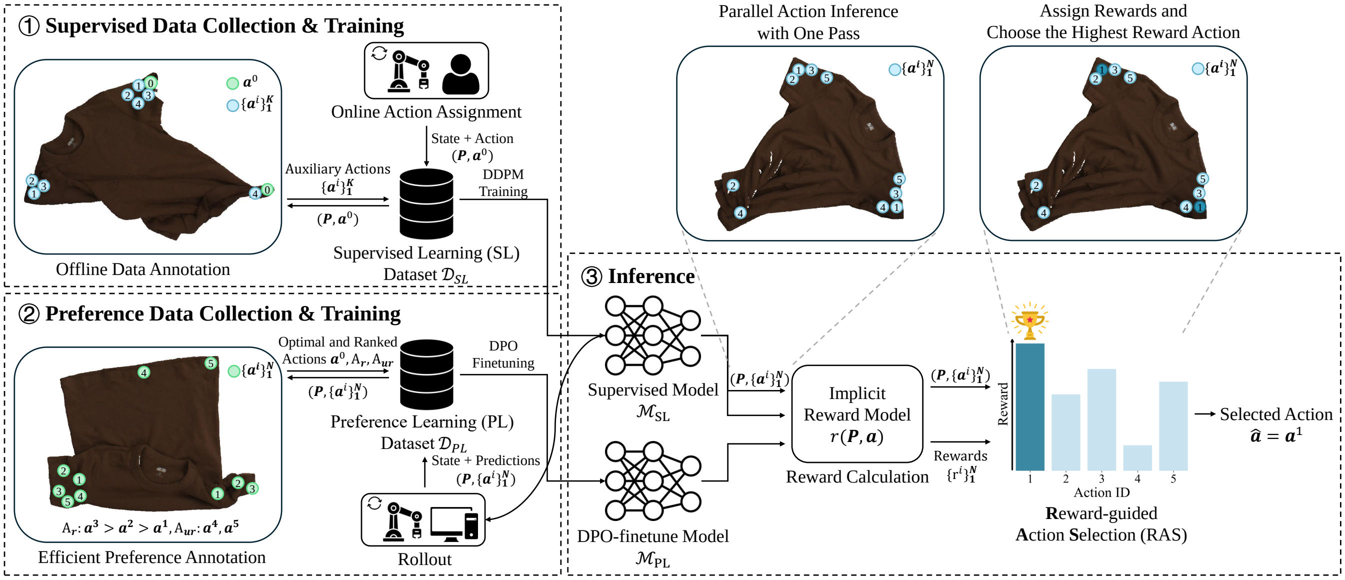


Fig. 2: Pipeline overview of **DeformPAM**. (1) In **stage 1**, we assign actions for execution and annotate auxiliary actions for supervised learning in a real-world environment and train a supervised primitive model based on Diffusion. Circles with the same numbers represent the manipulation positions for an action. (2) In **stage 2**, we deploy this model in the environment to collect preference data composed of annotated and predicted actions. These data are used to train a DPO-finetuned model. (3) During **inference**, we utilize the supervised model to predict actions and employ an implicit reward model derived from two models for **R**eward-guided **A**ction **S**election (**RAS**). The action with the highest reward is regarded as the final prediction.

IV. Methodology

We will illustrate our learning pipeline **DeformPAM** (see Fig. 2) in three parts: (1) In Section IV-A, we demonstrate how to train a diffusion-based primitive policy model with supervised learning. (2) In Section IV-B, we describe how to use preference data to learn an implicit reward model with DPO finetuning. (3) In Section IV-C, we present **R**eward-guided **A**ction **S**election (RAS), which boosts the performance of the supervised model during inference by using an implicit reward model to guide action selection.

A. Supervised Learning for an Initial Primitive Policy

We will first introduce the basics of action primitive learning in Sec. IV-A.1. Then we will illustrate how to collect data to train an initial primitive policy model with supervised learning in Sec. IV-A.2 and Sec. IV-A.3.

1) Action Primitive Learning: In order to improve data efficiency, we decompose long-horizon tasks into multiple action primitives, and our model predicts only the action parameters for each primitive. This approach not only reduces the horizon length [24] but also allows us to perform highly dynamic actions (e.g. fling a garment [13]). Our primitive learning network takes an RGB-D image $\mathcal{I}$ as input in each manipulation step. After that, Grounded SAM [39] is used to segment the point cloud $\mathbf{P}_t$ of the target object, then the policy model $\mathcal{M}$ will predict the predefined primitive action $\hat{\mathbf{a}} = \mathcal{M}(\mathbf{P})$. We use OMPL [40] to generate planning trajectories based on primitive parameters. PyBullet[41] and rule-based criteria are employed to ensure safety.

2) Data Collection: In order to acquire demonstration data, we design a graphic interface to let the user annotate one optimal action primitive parameter $\mathbf{a}^0$ for each observation step and let the robot execute the action primitive. Since the optimal actions in deformable objects manipulation are often diverse (multi-modal), we offline annotate additional K potentially optimal actions $\{\mathbf{a}^k\}_1^K$ called auxiliary actions (see Fig. 2 upper left) for previous seen observation states. Intuitively, using auxiliary actions allows the policy model to better understand the multi-modal nature of expert action distributions. These pairs of point cloud and action constitute the supervised learning dataset $\mathcal{D}_{SL}$.

3) Network Architecture: Our network takes a masked 3D point cloud as input. We adopt a ResUNet3D [42] and a lightweight Transformer [43] as backbone. We use a diffusion head to predict final action primitive parameters. To facilitate the efficiency of training and inference with auxiliary actions, we design a special technique for parallel training and inference. We reorganize the data in self-attention layers of the Transformer to prevent information leakage between distinct action tokens. This allows our network to simultaneously take multiple auxiliary actions and diverse noise in parallel for each state during training. Furthermore, during inference, for each state, our model can output multiple (N) potential actions in parallel with only one pass. We use the following DDPM [9] loss function for supervised learning:

$$L_{SL} = \mathbb{E}_{(\mathbf{a}_0, \mathbf{P}) \in \mathcal{D}_{SL}, t, \epsilon} \|\epsilon - \epsilon_\theta(\mathbf{a}_t, \mathbf{P}, t)\|_2^2. \tag{5}$$

B. Preference Learning by DPO Finetuning

To alleviate distribution shifts in long-horizon tasks, we collect a new round of on-policy data by running rollouts with the supervised model trained in Sec. IV-A. We annotate the data with human preferences, then use DPO [11] to finetune the supervised model. Next, we will describe how we collect preference data and illustrate the learning algorithm.

1) Data Collection: When we run rollouts with the pre-trained supervised model, we will record N predicted potential actions $\mathbf{A} = \{\mathbf{a}\}_1^N$ for each given state in one single pass. Annotators will first annotate an optimal action $\mathbf{a}^0$ then do the comparisons between these N predicted actions. Because N may be large, we design an efficient ranking-based preference data annotation strategy (see Fig. 2 lower left). During annotation, since some poor actions cannot be compared, annotators divide these actions into two groups: the better, rankable ones $\mathbf{A}_r$ and the poorer, unrankable ones $\mathbf{A}_{ur}$. Then actions in $\mathbf{A}$ are sorted and the preference data are generated by performing the Cartesian product among or between these groups, which is expressed as

$$\{(\mathbf{a}^w, \mathbf{a}^l)|\mathbf{a}^w, \mathbf{a}^l \in \mathbf{A}^r, \mathbf{a}^w \succ \mathbf{a}^l\} \cup \mathbf{A}_r \times \mathbf{A}_{ur} \cup \{\mathbf{a}^0\} \times \mathbf{A}. \quad (6)$$

Here, $\mathbf{a}^w \succ \mathbf{a}^l$ denotes action $\mathbf{a}^w$ win over action $\mathbf{a}^l$ in the ranking. Through this, we construct the preference learning dataset $\mathcal{D}_{PL}$. To enhance data efficiency, we prioritize using the more distant sample pairs in the sorted sequence during training.

2) Learning Algorithm: Once we have the preference dataset, we can finetune the policy model from a perspective similar to RLHF [33]. The RLHF objective maximizes a reward model $r(\mathbf{a}, \mathbf{P})$ while regularizing the difference with the initial reference model. Meanwhile, from the Bradley-Terry model [44], we also have another relation for $(\mathbf{a}^w, \mathbf{a}^l, \mathbf{P}) \in \mathcal{D}_{PL}$, which is

$$p(\mathbf{a}^w \succ \mathbf{a}^l|\mathbf{P}) = \sigma(r(\mathbf{a}^w, \mathbf{P}) - r(\mathbf{a}^l, \mathbf{P})). \quad (7)$$

Following DPO [10], we can indirectly train the RLHF objective by maximizing preference probability, bringing the policy closer to the optimal strategy under an implicit reward function. When the policy is a diffusion model, Diffusion-DPO [11] provides a corresponding loss function as

$$\begin{aligned} L_{PL} = &-\mathbb{E}_{(\mathbf{a}_0^w, \mathbf{a}_0^l, \mathbf{P}) \in \mathcal{D}_{PL}, t, \epsilon} \log \sigma \\ &\{-\beta T[(\|\epsilon - \epsilon_\theta(\mathbf{a}_t^w, \mathbf{P}, t)\|_2^2 - \|\epsilon - \epsilon_{SL}(\mathbf{a}_t^w, \mathbf{P}, t)\|_2^2) - \\ &(\|\epsilon - \epsilon_\theta(\mathbf{a}_t^l, \mathbf{P}, t)\|_2^2 - \|\epsilon - \epsilon_{SL}(\mathbf{a}_t^l, \mathbf{P}, t)\|_2^2)]\} \end{aligned} \quad (8)$$

where β is a regularization coefficient. This objective can be intuitively seen as encouraging denoising to $\mathbf{a}_0^w$ and penalizing denoising to $\mathbf{a}_0^l$, while trying to keep the finetuned model's predictions close to the pre-trained model's.

C. Parallel Inference with ***R**eward-guided **A**ction **S**election*

Preference learning enables models to differentiate between *good* and *bad* actions. However, with limited data, DPO finetuning can cause significant forgetting and performance degradation, a phenomenon observed in [45]. To reduce abnormal actions and alleviate distribution shifts, we propose **R**eward-guided **A**ction **S**election (RAS) to choose from the multiple actions predicted by the supervised model trained in IV-A (see Fig. 2 right).

A key byproduct of DPO finetuning is the implicit reward function. We exploit this to ensure robust action selection during inference. For the N potential actions predicted by the supervised model, we calculate the corresponding rewards and use a greedy strategy to select the action with the highest reward for execution. This inference process can be formulated as

$$\hat{\mathbf{a}} = \arg\max_{\mathbf{a} \in \mathcal{M}(\mathbf{P})} r(\mathbf{a}, \mathbf{P}) \quad (9)$$

where r is the reward function. As in Diffusion-DPO [11], we can compute r as

$$r(\mathbf{a}_0, \mathbf{P}) = -\mathbb{E}_{t,\epsilon} \beta T(\|\epsilon - \epsilon_{PL}(\mathbf{a}_t, \mathbf{P}, t)\|_2^2 - \|\epsilon - \epsilon_{SL}(\mathbf{a}_t, \mathbf{P}, t)\|_2^2). \quad (10)$$

It can be intuitively interpreted as evaluating the finetuned model's tendency of denoising to $\mathbf{a}_0$ while using the supervised model as a reference point.

To calculate rewards, we approximate the expectation through sampling. We observe that sampled values vary significantly across different diffusion timesteps t, with larger t producing smaller values. Thus, we use only the smallest 10% of timesteps for efficient reward calculation.

*How to Understand **R**eward-guided **A**ction **S**election (RAS)?* In the context of generative probabilistic policy models, such as diffusion models, the predicted actions typically form a multimodal distribution, concentrating around several centroids. In our experiments, we observe that for previously unseen states, the optimal action is often included among the multiple predictions generated by the policy model. However, the relative probability of this optimal action is generally low, resulting in its infrequent sampling. RAS can be understood as maintaining the original distribution of centroids while adjusting the assessment of their quality through reward-guided action selection. When online data are limited, the discriminative quality prediction can be generalized more effectively and efficiently to unseen states.

V. Experiments

We conduct experiments on three challenging real-world long-horizon manipulation tasks. We first describe the experimental design and baseline methods, then focus on examining **how does the model perform** and **what enables its capabilities** through quantitative and qualitative evaluations.

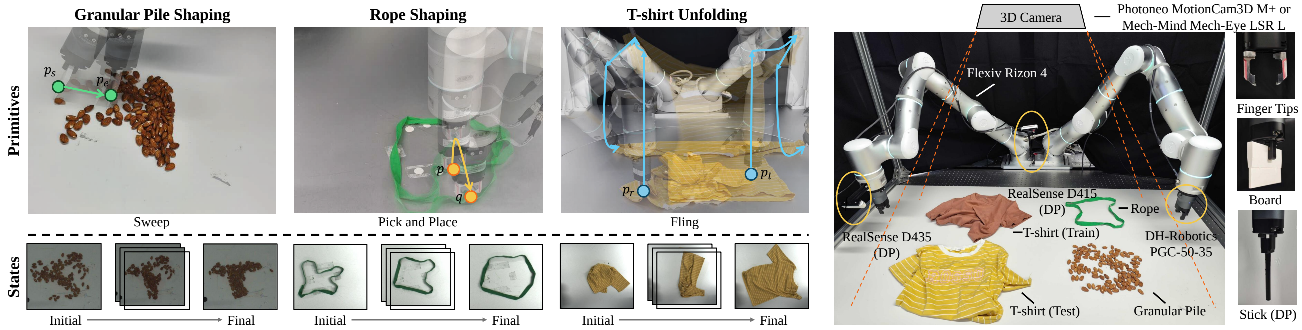
A. Tasks and Hardware Setup

As shown in Fig. 3a, we have designed three challenging long-horizon tasks: granular pile shaping, rope shaping and T-shirt unfolding. These tasks involve 1D, 2D, and granular deformable objects and all start with complex initial states. Next, we describe the definition of each task.

- **Granular Pile Shaping**: In this task, the robot sweeps a disordered pile of granular objects (*i.e.*, nuts) into the shape of the character T. We design a 3D-printed flat board as the robot tool and define the primitive parameters as $\mathbf{a} = (p_s, p_e)$, where p_s and p_s represent the start and end positions.
- **Rope Shaping**: In this task, the robot shapes a looped rope from a random shape into a circle using the pick-and-place primitive action $\mathbf{a} = (p, q)$, where p and q stand for the pick and place positions.
- **T-shirt Unfolding**: The goal of this task is to smooth out a short-sleeved T-shirt from a highly crumpled state. We use the fling action in Flingbot [13] as the primitive $\mathbf{a} = (p_l, p_r)$, where p_l and p_r denote the left and right pick positions.

We employ intersection over union (IoU), coverage, and Earth Mover's Distance (EMD) calculated between the current state and the target state to evaluate the completion quality during the execution process.

For hardware setup, the dual-arm platform and tools illustrated in Fig. 3b are used to conduct all the experiments.



(a) (b)

Fig. 3: (a) Object states and primitives of each task. Beginning with a random complex state of an object, multiple steps of action primitives are performed to gradually achieve the target state. (b) Hardware setup and tools used in our real-world experiments. Devices and tools marked with DP are not used in primitive-based methods.

B. Baselines and Implementation Details

We design the following primitive-based methods for quantitative comparison.

- **SL**: supervised model trained by offline data of stage 1.
- **SL + SL**: supervised model trained with offline data of stage 1 and the on-policy data (only the optimal actions) of stage 2.
- **DPO [10] + Implicit RAS**: DPO-finetuned model in stage 2 with implicit RAS during inference.
- **SL + Explicit RAS [44]**: We implement an explicit reward model by adding a prediction head (similar to the action diffusion head) to the supervised pretrained network in stage 1. It is trained by directly optimizing Eq. 7 using preference data of stage 2. We use the pre-trained supervised model in stage 1 for sampling actions and conduct reward-guided action selection (RAS) by explicit reward prediction.
- **SL + Implicit RAS *i.e.*, DeformPAM (Ours)**.

We train for 2000 epochs for supervised learning and 200 epochs for preference learning. All methods predict (sample) $N = 8$ actions for each state during data collection and evaluation. We only capture object states before/after each action primitive for all primitive-based methods. We also implement Diffusion Policy (DP) [1] with teleoperation data (RGB inputs, 10 FPS) as a primitive-free method only for qualitative comparison due to very different hardware and task settings. We annotate $K = 9$ auxiliary actions for each state in the supervised dataset D_{SL}. The specific dataset sizes are shown in the Tab. I.

TABLE I: The dataset size for each task. PB and DP denote Primitive-Based methods and Diffusion Policy [1]. # seq. and # states indicate the number of task sequences and states.

	Granular Pile		Rope		T-shirt	
	# seq.	# states	# seq.	# states	# seq.	# states
PB (Stage 1)	~ 60	400	~ 30	200	~ 90	200
PB (Stage 2)	~ 25	200	~ 10	100	~ 50	146
DP	60	29807	50	9971	-	-

C. Quantitative Evaluations

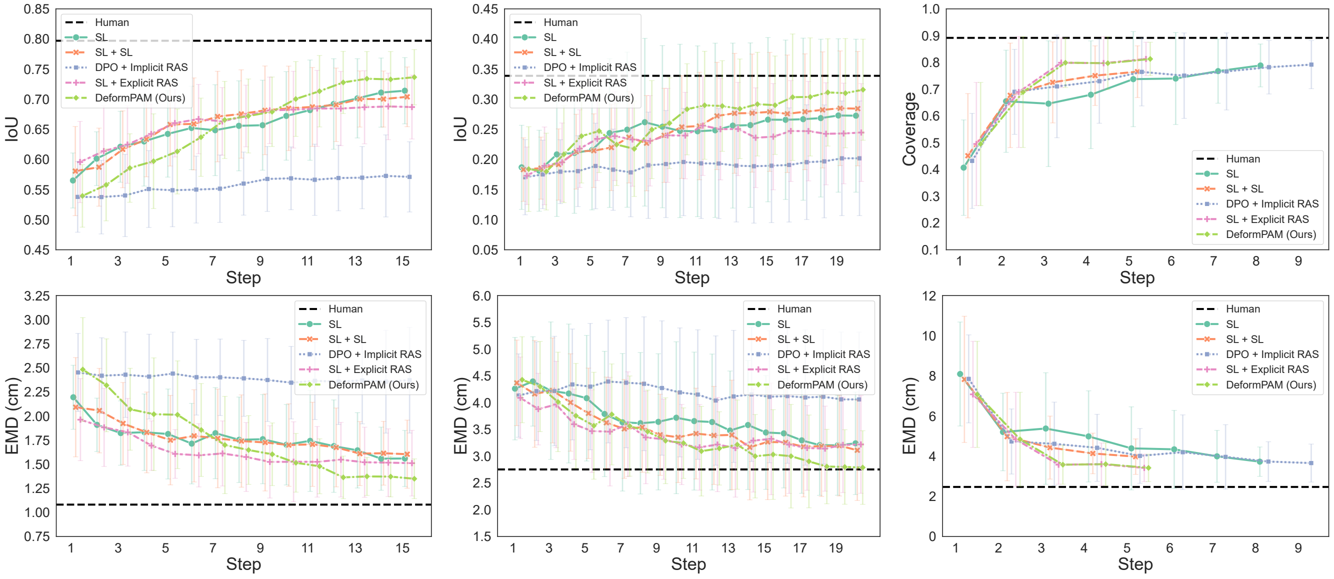
The quantitative metrics from the real-world experiments are presented in Fig. 4. We then illustrate the impacts of key components in our proposed framework and what enables its capabilities by answering the following questions.

Q1: Is using only supervised learning adequate for long-horizon tasks? As shown in Fig. 4, for the three tasks, with the help of reward-guided action selection, DeformPAM leads to an increase in the final completion quality. The variance in the quality metrics also tends to be smaller. Meanwhile, SL is more likely to generate abnormal action and get trapped in an intermediate state, preventing further improvement in the quality curve. The instability caused by these abnormal actions is mitigated through reward-guided action selection.

Q2: How about training a supervised model with both off-policy and on-policy data? Training with on-policy data is another method to alleviate distribution shifts. Although such a method can reduce the long-tail phenomenon of completion steps in Fig. 4c, the results in Fig. 4a and Fig. 4b indicate that SL + SL achieves only marginal improvements in harder tasks compared to the one using off-policy data. Thus, employing reward-guided action selection is a more efficient method to enhance model performance.

Q3: Does employing the finetuned model to predict action primitives result in better performance? As seen in Fig. 4a and Fig. 4b, DPO + Implicit RAS performs worse on the shaping tasks compared to the standard DeformPAM, and even underperforms the model using only supervised learning T-shirt Unfolding. It is probably due to the forgetting issues [45] in DPO finetuning, which leads to worse action prediction quality. This issue is more severe when data are limited, as is the case in this paper.

Q4: Is it more effective to extract the implicit reward model from DPO or to directly predict the reward? Besides extracting an implicit reward model, another way to obtain rewards is to directly train an explicit reward model with preference data. From Fig. 4a and Fig. 4b, it can be found that for harder tasks like shaping, it is challenging for SL + Explicit RAS to achieve a high completion quality as the standard DeformPAM. This may be caused by reward overfitting when the size of the preference dataset is limited. In contrast, an implicit reward model from the DPO-finetuned model can fully leverage the action distribu-

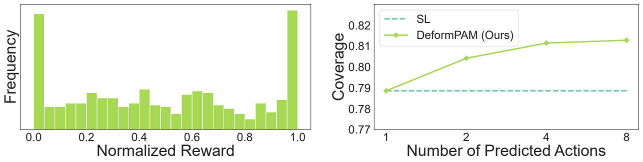


(a) Granular Pile Shaping

(b) Rope Shaping

(c) T-shirt Unfolding

Fig. 4: Quality metrics per step on the three tasks. The results are calculated on 20 trials. Each trial ends until the policy already reaches its optimal state or exceeds the maximum steps. SL, DPO, RAS stand for the supervised model, DPO-finetuned model, and reward-guided action selection.



(a)

(b)

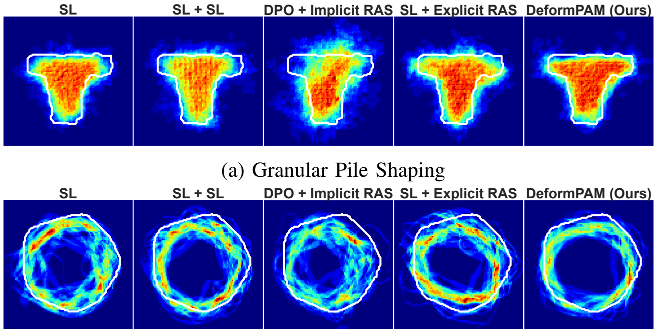
Fig. 5: (a) Normalized reward distribution during inference when sampling $N = 8$ actions. (b) Average coverage for various numbers N of predicted actions during inference.

tion learned during supervised learning. This phenomenon is inconsistent with conclusions in NLP [46], primarily because both pre-training and preference fine-tuning data are relatively abundant in NLP tasks. Actually, as in Fig. 4c, an explicit reward model can also achieve a good performance in a simpler task (*i.e.*, T-shirt unfolding) with more data.

Q5: How does reward-guided action selection (RAS) contribute to performance? We analyze the distribution of normalized implicit reward values during inference, as shown in Fig. 5a. This indicates that there is no positive correlation between the sampling probability of the action generation model and the predicted reward values, which suggests that employing RAS can serve as a quality reassessment. From another perspective, we compare the performance between random sampling and reward-guided action selection by adjusting the number N of predicted actions during inference in the T-shirt unfolding task and computing the final coverage. As shown in Fig. 5b, as N increases, the model's performance gradually improves. This demonstrates that RAS enables the model to select superior samples, thereby benefiting from a greater number of samples.

D. Qualitative Results

We draw the completion states of the granular pile shaping task and rope shaping task as heatmaps in Fig. 6. It is shown that our method achieves superior completion quality and exhibits lower variance. We also find that primitive-free methods like Diffusion Policy [1] easily get stuck in unseen states with limited data. For more detailed results, please refer to the supplementary video and the project website.



(a) Granular Pile Shaping

(b) Rope Shaping

Fig. 6: Final-state heatmaps compared with the target states.

VI. Conclusion

In this paper, we introduce DeformPAM, a novel framework for long-horizon deformable object manipulation that leverages preference-based action alignment to mitigate distributional shifts and enhance task performance. By integrating supervised learning with a preference learning model, DeformPAM employs reward-guided action selection to improve decision-making. Our experiments on three challenging real-world tasks demonstrate that DeformPAM enhances both task completion quality and efficiency compared to baseline methods. Future works could explore extending this approach to more complex tasks with multiple primitives.

Acknowledgements

This work is supported by the Shanghai Committee of Science and Technology, China (Grant No. 24511103200) by the National Key Research and Development Project of China (No. 2022ZD0160102), Shanghai Artificial Intelligence Laboratory, XPLORER PRIZE grants.

References

[1] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” in *Proceedings of Robotics: Science and Systems (RSS)*, 2023. 1, 5, 6

[2] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, “Learning fine-grained bimanual manipulation with low-cost hardware,” *arXiv preprint arXiv:2304.13705*, 2023.

[3] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, “3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations,” in *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[4] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song, “Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots,” in *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[5] T. Z. Zhao, J. Tompson, D. Driess, P. Florence, S. K. S. Ghasemipour, C. Finn, and A. Wahid, “ALOHA unleashed: A simple recipe for robot dexterity,” in *8th Annual Conference on Robot Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=gvdXE7ikHI 1

[6] S. Ross, G. Gordon, and D. Bagnell, “A reduction of imitation learning and structured prediction to no-regret online learning,” in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635. 1

[7] A. Canberk, C. Chi, H. Ha, B. Burchfiel, E. Cousineau, S. Feng, and S. Song, “Cloth funnels: Canonicalized-alignment for multi-purpose garment manipulation,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5872–5879. 1, 2

[8] Y. Avigal, L. Berscheid, T. Asfour, T. Kröger, and K. Goldberg, “Speedfolding: Learning efficient bimanual folding of garments,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 1–8. 1, 2

[9] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020. 1, 2, 3

[10] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, “Direct preference optimization: Your language model is secretly a reward model,” *Advances in Neural Information Processing Systems*, vol. 36, 2024. 1, 2, 4, 5

[11] B. Wallace, M. Dang, R. Rafailov, L. Zhou, A. Lou, S. Purushwalkam, S. Ermon, C. Xiong, S. Joty, and N. Naik, “Diffusion model alignment using,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 8228–8238. 1, 2, 3, 4

[12] H. Xue, Y. Li, W. Xu, H. Li, D. Zheng, and C. Lu, “Unifolding: Towards sample-efficient, scalable, and generalizable robotic garment folding,” in *Conference on Robot Learning*. PMLR, 2023, pp. 3321–3341. 2

[13] H. Ha and S. Song, “Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding,” in *Conference on Robot Learning*. PMLR, 2022, pp. 24–33. 3, 4

[14] R. Wu, H. Lu, Y. Wang, Y. Wang, and H. Dong, “Unigarmentmanip: A unified framework for category-level garment manipulation via dense visual correspondence,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 16 340–16 350.

[15] Z. Xu, Z. Xian, X. Lin, C. Chi, Z. Huang, C. Gan, and S. Song, “Roboninja: Learning an adaptive cutting policy for multi-material objects,” *arXiv preprint arXiv:2302.11553*, 2023. 2

[16] Y. Wang, Z. Sun, Z. Erickson, and D. Held, “One policy to dress them all: Learning to dress people with diverse poses and garments,” in *Robotics: Science and Systems (RSS)*, 2023. 2

[17] L. Y. Chen, B. Shi, D. Seita, R. Cheng, T. Kollar, D. Held, and K. Goldberg, “Autobag: Learning to open plastic bags and insert objects,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 3918–3925. 2

[18] S. Chen, Y. Xu, C. Yu, L. Li, and D. Hsu, “Differentiable particles for general-purpose deformable object manipulation,” *arXiv preprint arXiv:2405.01044*, 2024. 2

[19] K. Zhang, B. Li, K. Hauser, and Y. Li, “Adaptigraph: Material-adaptive graph-based neural dynamics for robotic manipulation,” in *Proceedings of Robotics: Science and Systems (RSS)*, 2024. 2

[20] A. Mandlekar, F. Ramos, B. Boots, S. Savarese, L. Fei-Fei, A. Garg, and D. Fox, “Iris: Implicit reinforcement without interaction at scale for learning control from offline robot manipulation data,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4414–4420. 2

[21] A. Mandlekar, D. Xu, R. Martín-Martín, S. Savarese, and L. Fei-Fei, “Learning to generalize across long-horizon tasks from human demonstrations,” *arXiv preprint arXiv:2003.06085*, 2020.

[22] K. Shiarlis, M. Wulfmeier, S. Salter, S. Whiteson, and I. Posner, “Taco: Learning task decomposition via temporal alignment for control,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 4654–4663.

[23] D. Xu, S. Nair, Y. Zhu, J. Gao, A. Garg, L. Fei-Fei, and S. Savarese, “Neural task programming: Learning to generalize across hierarchical tasks,” in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 3795–3802.

[24] T. Gao, S. Nasiriany, H. Liu, Q. Yang, and Y. Zhu, “Prime: Scaffolding manipulation tasks with behavior primitives for data-efficient imitation learning,” *arXiv preprint arXiv:2403.00929*, 2024. 2, 3

[25] C. Wang, L. Fan, J. Sun, R. Zhang, L. Fei-Fei, D. Xu, Y. Zhu, and A. Anandkumar, “Mimicplay: Long-horizon imitation learning by watching human play,” in *Conference on Robot Learning*. PMLR, 2023, pp. 201–221. 2

[26] C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet, “Learning latent plans from play,” in *Conference on robot learning*. PMLR, 2020, pp. 1113–1132.

[27] Z. J. Cui, Y. Wang, N. M. M. Shafiullah, and L. Pinto, “From play to policy: Conditional behavior generation from uncurated robot data,” in *The Eleventh International Conference on Learning Representations*.

[28] E. Rosete-Beas, O. Mees, G. Kalweit, J. Boedecker, and W. Burgard, “Latent plans for task-agnostic offline reinforcement learning,” in *Conference on Robot Learning*. PMLR, 2023, pp. 1838–1849. 2

[29] H. Shi, H. Xu, S. Clarke, Y. Li, and J. Wu, “Robocook: Long-horizon elasto-plastic object manipulation with diverse tools,” in *Conference on Robot Learning*. PMLR, 2023, pp. 642–660. 2

[30] D. Sadigh, A. Dragan, S. Sastry, and S. Seshia, *Active preference-based learning of reward functions*, 2017. 2

[31] B. Ibarz, J. Leike, T. Pohlen, G. Irving, S. Legg, and D. Amodei, “Reward learning from human preferences and demonstrations in atari,” *Advances in neural information processing systems*, vol. 31, 2018.

[32] E. Bıyık, N. Huynh, M. J. Kochenderfer, and D. Sadigh, “Active preference-based gaussian process regression for reward learning,” *arXiv preprint arXiv:2005.02575*, 2020. 2

[33] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, “Deep reinforcement learning from human preferences,” *Advances in neural information processing systems*,

vol. 30, 2017. 2, 4

[34] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022. 2

[35] J. Hejna, R. Rafailov, H. Sikchi, C. Finn, S. Niekum, W. B. Knox, and D. Sadigh, "Contrastive prefence learning: Learning from human feedback without rl," *arXiv preprint arXiv:2310.13639*, 2023. 2

[36] M. Kim, Y. Lee, S. Kang, J. Oh, S. Chong, and S. Yun, "Preference alignment with flow matching," *arXiv preprint arXiv:2405.19806*, 2024. 2

[37] A. Askell, Y. Bai, A. Chen, D. Drain, D. Ganguli, T. Henighan, A. Jones, N. Joseph, B. Mann, N. DasSarma *et al.*, "A general language assistant as a laboratory for alignment," *arXiv preprint arXiv:2112.00861*, 2021. 2

[38] M. Khanov, J. Burapacheep, and Y. Li, "Args: Alignment as reward-guided search," in *The Twelfth International Conference on Learning Representations*. 2

[39] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan *et al.*, "Grounded sam: Assembling open-world models for diverse visual tasks," *arXiv preprint arXiv:2401.14159*, 2024. 3

[40] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, 2012. 3

[41] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," 2016. 3

[42] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 8958–8966. 3

[43] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017. 3

[44] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: I. the method of paired comparisons," *Biometrika*, vol. 39, no. 3/4, pp. 324–345, 1952. 4, 5

[45] A. Pal, D. Karkhanis, S. Dooley, M. Roberts, S. Naidu, and C. White, "Smaug: Fixing failure modes of preference optimisation with dpo-positive," *arXiv preprint arXiv:2402.13228*, 2024. 4, 5

[46] Y. Lin, S. Seto, M. ter Hoeve, K. Metcalf, B.-J. Theobald, X. Wang, Y. Zhang, C. Huang, and T. Zhang, "On the limited generalization capability of the implicit reward model induced by," *arXiv preprint arXiv:2409.03650*, 2024. 6